Pergamon

Appl. Math. Lett. Vol. 11, No. 6, pp. 87–91, 1998

Printed in Great Britain
0893-9659/98 $19.00 + 0.00

PII: S0893-9659(98)00108-6

Calculating the Normalization Constants in Queueing Networks

H. YANG
Pivotal Networking, Inc.
7246 Sharon Drive, Suite P
San Jose, CA 95129, U.S.A.
hyang@pivnet.com

W.-B. GONG
Department of Electrical and Computer Engineering
University of Massachusetts at Amherst
Amherst, MA 01003, U.S.A.
gong@ecs.umass.edu

(Received May 1995; accepted October 1997)

Abstract—Many queueing network models have a product-form solution for their steady-state probability distributions. However, the calculation of the normalization constants involved in the solutions is nontrivial. Recently, Gordon proposed a method to derive the closed form formula for normalization constants for certain closed networks. In this paper, we describe a simpler method using Z-transform. In the cases of networks of multiple server queues or of single server queues with equal traffic intensities, the computation involved in proposed approach is much simpler than that in Gordon's paper. For multichain closed networks, we propose to use FFT (Fast-Fourier-Transform) to calculate the normalization constants. © 1998 Elsevier Science Ltd. All rights reserved.

Keywords—Queueing networks, Normalization constants, Z-transform, Product form networks, Multichain networks.

1. INTRODUCTION

Queueing networks of the product-form type are widely used as models in the analysis and design of, for example, computer and telecommunication systems. But for many systems (closed networks or others with population constraints), the product-form solution involves a normalization constant, whose calculation is not trivial, and there is much research work done in the literature (see, e.g., [1], for surveys).

The origin of computational algorithms for normalization constants may be attributed to Buzen. In 1973, Buzen [2] presented a convolution algorithm for single chain closed queueing networks. Reiser and Kobayashi [3] extended this convolution algorithm to multichain queueing networks. The problem with the convolution algorithm is that the computation costs increases quickly with the system size (exponentially in number of classes, polynomially in population). Harrison [4] first obtained some closed form expression for normalization constants. The method was later on elaborated by Gordon [5].

Part of this paper has been presented in the IEEE Conference of Decision and Control, 1994 (see [6]).

Typeset by $\mathcal{AMS}$-TEX

We propose an alternative method here also to obtain the closed form formula for normalization constants. This approach is based on the fact that calculation of these normalization constants involves convolution procedures, and therefore uses the Z-transform. Our treatment is simpler and more straightforward than that in [5].

The rest of the paper is organized as follows. In Section 2, we describe the models and the basic idea of the method. In Section 3, several examples are provided. We also propose to use FFT (Fast-Fourier-Transform) to calculate the normalization constants in multichain closed queueing networks.

2. NORMALIZATION CONSTANTS IN PRODUCT FORM CLOSED NETWORKS

In this section, we consider the single-class product-form closed queueing network of load-dependent queues. There are M queues and a total of N customers in the network. The service time at the i^{th} queue when there are a total of n_i customers present at that queue is exponentially distributed with mean $1/(a_i(n_i) \cdot \mu_i)$. The routing is probabilistic, namely, a customer goes to queue j after service in queue i with probability p_{ij}.

This queueing network has a product-form solution for its steady-state probability distribution (see [2]). Let $(n_1, n_2, \dots, n_M)$ be the state vector. Let x_i, $i = 1, 2, \dots, M$, be the "visiting ratios" (or relative traffic intensities) obtained by solving

$$x_j \mu_j = \sum_{i=1}^{M} x_i \mu_i p_{ij}, \qquad 1 \le j \le M.$$

Then the queue length distribution (see [2])

$$P(n_1, n_2, \dots, n_M) = \frac{1}{G(N)} \prod_{i=1}^{M} \frac{x_i^{n_i}}{A_i(n_i)},$$

where

$$A_i(k) = \begin{cases} \prod_{j=1}^{k} a_i(j), & \text{if } k > 0, \\ 1, & \text{if } k = 0. \end{cases}$$

The normalization constant

$$G(N) = \sum_{S(N,M)} \prod_{i=1}^{M} \frac{x_i^{n_i}}{A_i(n_i)},$$

where the summation is taken over the state space $S(N, M)$.

Define

$$g(n, m) = \sum_{S(n,m)} \prod_{i=1}^{m} \frac{x_i^{n_i}}{A_i(n_i)}.$$

Then $G(N) = g(N, M)$.

The convolution algorithm calculates $g(n, m)$ iteratively (see [2])

$$g(n, m) = \sum_{k=0}^{n} \frac{x_m^k}{A_m(k)} g(n - k, m - 1).$$

The Z-transform method is based on the above convolution equation. If for each node i, $i = 1, 2, \dots, M$, we define a sequence $\{X_i(k) = (x_i^k / A_i(k)),\ k = 0, 1, \dots\}$, then

$$\begin{aligned} g(n, m) &= X_m(n) \otimes g(n, m - 1) \\ &= X_1(n) \otimes X_2(n) \otimes \cdots \otimes X_m(n) \otimes g(n, 0), \end{aligned}$$

where $g(0,0) = 1$, $g(n,0) = 0$, $n = 1,2,\dots$, and $\otimes$ denotes the convolution operation of two sequences.

Taking Z-transform, we obtain

$$G(z,M) = X_1(z)X_2(z)\dots X_M(z),$$

where $G(z,M)$ is the Z-transform of $\{g(n,M), n = 0,1,\dots\}$, and $X_i(z)$ is the Z-transform of $\{X_i(n), n = 0,1,\dots\}$. Note that $Z\{g(n,0)\} = 1$.

Finally,

$$g(n,M) = Z^{-1}\left\{X_1(z)X_2(z)\dots X_M(z)\right\}.$$

3. EXAMPLES OF CLOSED FORM FORMULA OF NORMALIZATION CONSTANTS

3.1. Network of Single Server Queues

If each queue in the network contains a single server with the service rate independent of system state, i.e.,

$$a_i(n_i) = 1, \qquad i = 1,2,\dots,M,$$

then

$$X_i(n) = x_i^n.$$

We have

$$X_i(z) = Z\{X_i(n)\} = \frac{1}{1 - x_i z^{-1}}.$$

Then

$$g(n,M) = Z^{-1}\left\{\prod_{i=1}^{M} \frac{1}{1 - x_i z^{-1}}\right\}.$$

If we assume all traffic intensities are distinct, i.e., $x_i \neq x_j$, for $i \neq j$, then taking inverse Z-transform of the above equation yields

$$g(n,M) = \sum_{i=1}^{M} \frac{x_i^{n+M-1}}{\prod_{j\neq i}(x_i - x_j)}.$$

This closed formula was first obtained by Harrison [4]. Gordon [5] first proposed a method to derive it. Gordon's method is successful, but it introduced a "dummy variable" and a delta function, used contour integral as well. Compared to this, the above method is simpler and more straightforward.

When there are equal traffic intensities (e.g., $x_i = x_j$, for some $i,j \in 1,2,\dots,M$), Gordon's method becomes more involved, while the Z-transform approach remains simple—just do the inverse Z-transform, by calculating residues.

3.2. Network of Multiserver Queues

Consider a multiserver queueing network. At node i, there are K_i servers ($i = 1,2,\dots,M$) and each server has the same service rate μ_i. In this case,

$$a_i(n_i) = \begin{cases} n_i, & \text{for } 0 \le n_i < K_i, \\ K_i, & \text{for } n_i \ge K_i. \end{cases}$$

We then have

$$X_i(n) = \begin{cases} \dfrac{x_i^n}{n!}, & \text{for } 0 \le n < K_i, \\ \dfrac{x_i^n}{K_i! K_i^{n_i - K_i}}, & \text{for } n \ge K_i. \end{cases}$$

Taking Z-transform, we obtain

$$X_i(z) = \sum_{j=0}^{K_{i-1}} \frac{(x_i/z)^j}{j!} + \frac{x_i^{K_i}}{(z^{K_i})\,K_i!}\frac{1}{1-x_i z^{-1}}.$$

The closed form normalization constants can be obtained as

$$g(n,M) = Z^{-1}\left\{\prod_{i=1}^{M}\left[\sum_{j=0}^{K_{i-1}} \frac{(x_i/z)^j}{j!} + \frac{x_i^{K_i}}{(z^{K_i})\,K_i!}\frac{1}{1-x_i z^{-1}}\right]\right\}.$$

Note that here we have a rational function of z, and therefore, the inverse Z-transform can be performed easily by calculating residues, e.g., using some symbolic operation software such as MATHEMATICA (see, e.g., [7]). The computation involved here is much simpler than the corresponding algorithm given in [5], which is a recursive algorithm iterating on the number of servers at each node.

Consider another example as an extreme case of multiserver: all nodes contain infinite number of servers. Now

$$a_i(n) = n, \qquad \text{for } n = 0,1,2,\ldots.$$

We have

$$X_i(n) = \frac{x_i}{n!}, \qquad \text{for } n = 0,1,2,\ldots.$$

Then

$$X_i(z) = Z\left\{\frac{x_i}{n!}\right\} = e^{x_i/z}.$$

The normalization constants is

$$g(n,M) = Z^{-1}\left\{e^{(x_1+x_2+\cdots+x_M)/z}\right\} = \frac{(x_1+x_2+\cdots+x_M)^n}{n!},$$

which has been obtained by Gordon [5].

3.3. Multiple-Chain Queueing Network

Multiple-chain closed queueing networks are more important than single chain ones in modeling and analysis. The calculation of normalization constants in this case becomes a multi-dimensional convolution (see [8]). We can again apply Z-transform, but since there is no general method to perform multi-dimensional inverse Z-transform, we are not likely to obtain closed-form expressions.

Consider a simplest example of a two-node two-chain product-form closed queueing network. Suppose the relative traffic intensities at two nodes for the first class of customers are 1, a, respectively, and those for the second class of customers are 1, b, respectively.

Let $g(n_1,n_2) = g_2(n_1,n_2)$ be the normalization constant for two-node system with population size (n_1,n_2), and $g_1(n_1,n_2)$ for the system with the first node.

$$g_2(n_1,n_2) = \sum_{i=0}^{n_1}\sum_{j=0}^{n_2} p_2(i,j)\,g_1(n_1-i,n_2-j),$$
$$g_1(n_1,n_2) = \sum_{i=0}^{n_1}\sum_{j=0}^{n_2} p_1(i,j)\,g_0(n_1-i,n_2-j),$$

where $g_0(0,0)=1$, $g_0(i,j)=0$ (if $i\neq 0$ or $j\neq 0$), and (see [8])

$$p_1(i,j) = \frac{(i+j)!}{i!j!},$$
$$p_2(i,j) = \frac{(i+j)!a^i b^j}{i!j!}.$$

Then,

$$g(n_1, n_2) = p_1(n_1, n_2) \otimes p_2(n_1, n_2) \otimes g_0(n_1, n_2).$$

Take two-dimensional Z-transform (see, e.g., [9])

$$G(z_1, z_2) = P_1(z_1, z_2) P_2(z_1, z_2),$$

where

$$P_1(z_1, z_2) = Z\{p_1(n_1, n_2)\} = \frac{1}{1 - z_1^{-1} - z_2^{-1}},$$
$$P_2(z_1, z_2) = Z\{p_2(n_1, n_2)\} = \frac{1}{1 - a z_1^{-1} - b z_2^{-1}}.$$

We have

$$G(z_1, z_2) = \frac{1}{\left(1 - z_1^{-1} - z_2^{-1}\right)\left(1 - a z_1^{-1} - b z_2^{-1}\right)}.$$

Thus,

$$g(n_1, n_2) = Z^{-1}\left\{\frac{1}{\left(1 - z_1^{-1} - z_2^{-1}\right)\left(1 - a z_1^{-1} - b z_2^{-1}\right)}\right\}.$$

To the authors' knowledge, there is no closed-form expression for the above inverse two-dimensional Z-transform. This indicates that general closed-form formula for the normalization constant of multichain networks may not exist. In [10], asymptotic expansion for the normalization constants for multichain networks is obtained using Z-transform. Choudhury, Leung and Whitt [11] uses numerical inversion of multidimension Z-transform to calculate the normalization constants. It is also possible to use FFT to do the calculation. Consider a product-form closed network with M nodes, K customer chains (each with population N). Suppose the service rates of the nodes are load-dependent, then the computation time will be of the order of $(M-1)[(N+1)(N+2)/2]^K$ using the convolution algorithm (see [8]). If we perform multidimensional FFT to the convolution, the computation complexity will be reduced to $O\{M(N \log N)^K\}$.

REFERENCES

1. A.E. Conway and N.D. Georganas, *Queueing Networks—Exact Computational Algorithms*, The MIT Press, (1989).
2. J.P. Buzen, Computational algorithms for closed queueing networks with exponential servers, *Communications of ACM* **16**, 527–531, (1973).
3. M. Reiser and H. Kobayashi, Queueing networks with multiple closed chains: Theory and computational algorithms, *IBM J. Res. and Dev.* **19**, 283–294, (1975).
4. P.G. Harrison, On normalizing constants in queueing networks, *Opns. Res.* **33**, 464–468, (1985).
5. J.J. Gordon, The evaluation of normalizing constants in closed queueing networks, *Opns. Res.* **38**, 863–869, (1990).
6. H. Yang and W.B. Gong, On calculating the normalization constants in queueing networks, In *Proc. of the 33^{rd} CDC*, pp. 2075–2076, (1994).
7. S. Wolfram, *Mathematica, A System for Doing Mathematics by Computer*, Addison-Wesley, New York, (1991).
8. S.S. Lam and Y.L. Lien, A tree convolution algorithm for the solution of queueing networks, *Comm. of ACM* **26**, 203–215, (1983).
9. D.E. Dudgeon and R.M. Mersereau, *Multidimensional Digital Signal Processing*, Prentice Hall, (1984).
10. A. Birman and Y. Kogan, Asymptotic evaluation of closed queueing networks with many stations, *Commun. Statist.—Stochastic Models* **8** (3), 543–563, (1992).
11. G.L. Choudhury, K.K. Leung and W. Whitt, Numerical inversion of generating functions for multichain queueing networks, In *Proc. of 5^{th} IEEE International Workshop on Computer-Aided Modeling, Analysis, and Design*, Princeton, NJ, (1994).
12. A.C. Williams and R.A. Bhandiwad, A generating function approach to queueing network analysis of multiprogrammed computers, *Networks* **6** (1), 1–22, (1976).